CLAIMS

1   1. (Original)   A method for dynamic emulation of legacy instructions of a legacy program

2   comprising:

3   providing state information for determining a program execution mode for emulating said legacy

4   instructions,

5   accessing said legacy instructions and said state information,

6   for each particular legacy instruction,

7   querying to determine if one or more particular translated instructions for said execution

8   mode are stored as a result of translating said legacy instruction for said execution mode,

9   and

10   if not translated for said execution mode,

11   translating the particular legacy instruction into one or more particular translated

12   instructions for emulating the particular legacy instruction for said execution

13   mode,

14   storing said one or more particular translated instructions with said state

15   information, and

16   if translated for said execution mode, continuing without additional translating,

17   accessing said one or more particular translated instructions for emulating said legacy

18   instructions for said execution mode.

1   2. (Original)   The method of Claim 1 wherein said storing of the one or more particular translated

2   instructions is in one or more particular translated blocks and said state information is stored in each

3   of said particular translated blocks.

1   3. (Original)   The method of Claim 1 wherein said legacy instructions are for a legacy system having

2   a S/390 architecture.

1   4. (Original)   The method of Claim 1 wherein said legacy instructions are object code instructions

2   compiled/assembled for a legacy architecture.

1   5. (Original)   The method of Claim 1 wherein said translated instructions are for execution in a

2   RISC architecture.

1   6. (Original)   A method for dynamic emulation of legacy instructions, where the legacy instructions

2   are compiled/assembled into object code form for a native architecture, where the legacy instructions

3   are executed as guests in the host architecture, where the legacy instructions are translated to

4   translated instructions in the host architecture and the translated instructions are executed in the host

5   architecture concurrently with the translation of the legacy instructions in the host architecture,

6   comprising:

7   providing state information for determining a program execution mode for emulating said legacy

8   instructions,

9   accessing said legacy instructions and said state information as guests in the host architecture,

10   for each particular legacy instruction,

11   querying to determine if one or more particular translated instructions for said execution

12   mode are stored as a result of translating said legacy instruction for said execution mode,

13   and

14   if not translated for said execution mode,

15   translating the particular legacy instruction into one or more particular translated

16   instructions for emulating the particular legacy instruction for said execution

17   mode,

18   storing said one or more particular translated instructions with said state

19   information, and

20   if translated for said execution mode, continuing without additional translating,

21   accessing said one or more particular translated instructions for emulating said legacy

22   instructions for said execution mode as a guest in said host architecture.